

Call for Tender for Application Development and Maintenance Services

Annex 2 - Agile Application Development and
Maintenance

Appendix A - OECD's Agile Practices and
Guidelines

Contents

1. INTRODUCTION	2
2. CONTEXT	2
3. VALUES AND PRINCIPLES	2
3.1 GENERIC PRINCIPLES.....	3
4. FRAMEWORK OF PRACTICES	3
4.1 A SCRUM-BASED PROJECT LIFE CYCLE	3
4.1.1 <i>Global process</i>	3
4.1.2 <i>Estimations</i>	4
4.1.3 <i>Continuous improvement</i>	5
4.1.4 <i>Team structure and organization</i>	5
4.2 AGILE ROLES DEFINITIONS AT OECD	6
4.3 AGILE ARTIFACTS	6
4.4 INDICATORS.....	6
5. COMMON TOOLBOX	8
5.1 BACKLOG AND MANAGEMENT	8
5.2 CONTINUOUS INTEGRATION	8
5.3 OTHER COMMON TOOLS	8

1. Introduction

This paper proposes a synthesis view of current agile practices in the IT department at the OECD. The document is the result of several workshops and consultations with IT department team members that took place since September 2014. It describes the OECD state of work in term of Agile adoption by focusing on main shared principles, practices and tools.

2. Context

OECD has chosen to orient its project organization on agile methodologies for most of its internal IT projects, the motivations for this choice are:

- To preserve knowledge and skills of core business solutions inside OECD
- To be able to very quickly react to change
- To drive projects continuously by business value at all levels
- To be stay focused on end users requirements and feedback
- To promote innovation

The context for each of the OECD IT teams is different:

- Different by scope and nature of work to be realized: development, bug fix, integration, setup, installation...
- Different by the structure of team: different team size and localization
- Different stakeholders, business users and cultural differences: different business involvements, team history and individual background

This various contexts deeply influence projects organizations and processes.

Despite these differences, teams have developed common practices about project management and organization around agile methodologies.

Obviously the maturity level of each team is very various but the recent work on formalization and convergence enabled to identify **a common set of target practices and tools**. The global adoption of all this tools and practices is ongoing, each team evolve at his own rhythm according to their constraints and priorities.

The definition of this target framework will evolve continuously in a logic of continuous improvement.

3. Values and principles

Much more than adopting a strict agile methodology, the OECD teams adhere to core values of agility as defined in the agile manifesto¹ :

- **Individuals and interactions** rather than processes and tools
- **Operational functionality** rather than exhaustive documentation
- **Collaboration with the client** rather than a contractual relationship
- **Acceptance of change** rather than compliance with plans

The concrete translation of these values into actionable principles is variable according to the project context. But despite this variation, OECD has globally defined a target agile project framework for all teams. All common practices and tools described below are defining this target

¹ See the "Agile Manifesto for Agile Software Development" - <http://www.agilemanifesto.org/>

framework. The global adoption of this core framework is ongoing under the responsibility of each team.

3.1 Generic principles

- A Project time line structured by regular periodic meetings
- An iterative development cycle for building incremental solutions
- The Project efficiency relies on team member involvement
- An active collaboration with clients for gathering requirements and feedback management
- Regular Meta work for continuous improvement

All this principles are parts of the core of agile methodologies and can be implemented in several ways. The key points are to be focused and align with this principles much more than applying strictly the methodology.

Teams are often mixing internal resources and external collaborators, co-localized or not, this requires adaptations in project organizations.

By defining a target framework on the top of these principles OECD wants to be able to:

- Simplify the communication between teams
- Share practices and skills across teams
- Increase the global maturity level
- Be more readable for external actors
- Be more predictive and professional

4. Framework of practices

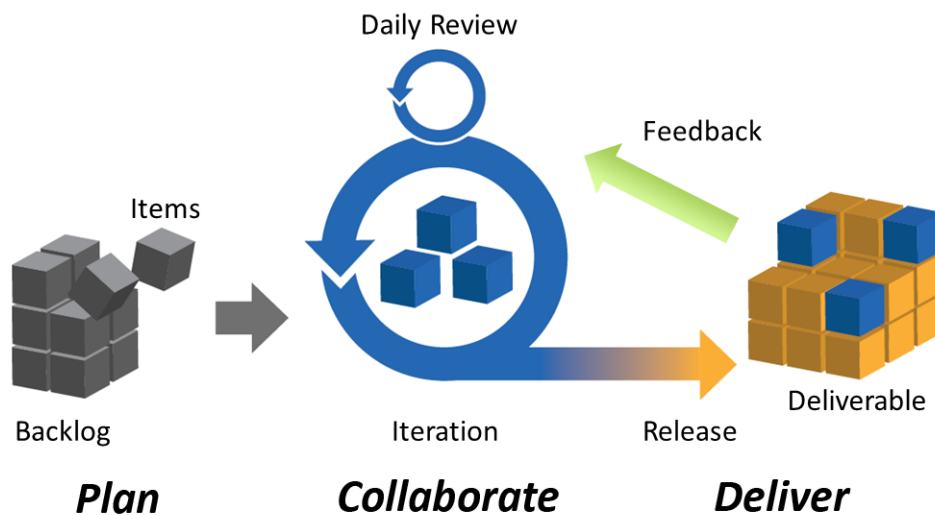
4.1 A Scrum-based project life cycle

4.1.1 Global process

Projects are organized following a Scrum cycle of delivery. A continuously prioritised product backlog is treated iteratively during “sprints” by a technical team.

At the end of each sprint:

- A quick demo is organized to collect end users feedback
- A retrospective takes place to evaluate and improve the overall process
- A new planning session is organized to define the content of the new sprint



- The length of the sprint iteration is now most frequently 2 weeks but can for certain projects be one month according to its nature and constraints.
- In some projects the workflow can be continuous without notion of sprint following kanban² principles.
- The organization of the different meetings is variable according to:
 - Team localisations and schedules
 - End users availabilities and involvement
- The product can be delivered in production at the end of each sprint or less often according to a release plan grouping several sprints.

4.1.2 Estimations

Each team maintains his relative estimation scale, it can be :

- Story points (most frequent)
- Tee-shirt size
- Hours
- Days

Each story is estimated during the planning session before being allocated to a sprint.

All team members have to participate to the estimation (based on their skills) using a collaborative method:

- Vote
- Poker planning
- others

² http://en.wikipedia.org/wiki/Kanban_%28development%29

4.1.3 Continuous improvement

The process of continuous improvement is under the responsibility of the team, conducted by the Scrum Master and organized around the retrospective ceremony.

The goal of this ceremony is:

- To celebrate success
- Analyse failures and difficulties
- Define SMART actions to experiment during the next iterations

4.1.4 Team structure and organization

Teams are globally following these rules:

- 7 +/-2 members
- Every member has Scrum skills and an agile sensibility
- Mixing people from IT and business in a daily collaboration
- Each team has at least 1 or 2 members from internal OECD staff. Other members can be external supplier's consultants (onsite or centralized offsite).

In case of distributed teams:

- The telepresence environment is fully operational: visio/tel/chat/large screen/etc.
- Periodic physical presence work time
- Distant "Proxy Scrum Master" for a distant team larger than one person

For maximal efficiency OECD tries to ensure the stability of teams and avoid excessive turnover, including for offsite resources. A process of on-boarding and knowledge transfer has nevertheless been defined to facilitate new resources integration and simplify transition:

1. Minimum knowledge transfer of 1 month between existing resource departure and new resource start date
2. Remote team to contribute to precise on-boarding procedure document to facilitate new resource knowledge transfer
3. Resource to be physically present onsite for up to 1 month within a maximum of 3 weeks of starting

Please see Appendix C of Annex 2 for the Onboarding Procedure for Agile Application Development and Maintenance.

4.2 Agile roles definitions at OECD

The role definition is mostly based on classical scrum roles with a few adaptations.

Product Owner	Scrum Team	ScrumMaster
<ul style="list-style-type: none">• Product vision• Prioritizes work to maximize ROI• Determines when to deploy• Keeps team "fed" with high value work	<ul style="list-style-type: none">• Self organizing• Cross functional skills• Creates and enforces own ground rules• Responsible for commitments	<ul style="list-style-type: none">• Works for the team• Coach, Leader, Facilitator, Change Agent• Removes impediments• Has no authority

The product owning is often shared between a "backlog master" and the project board:

- Project Board
 - Composed of decision maker
 - Feed the backlog manager with strategic vision
 - Validate results and priorities according to the vision
- Backlog master
 - Translate the vision into a scrum backlog by gathering detailed requirements from key users
 - Support the team by answering daily questions

In case of distributed team, a Proxy Scrum Master is required for a remote team of more than one member. This Proxy Scrum Master is supposed to have a role of remote facilitator and coordinator. He is responsible for:

- Collecting impediments
- Ensuring the correct affectation of tasks
- Regulating the remote team workload

4.3 Agile artifacts

There are two core artifacts:

- The product backlog, managed in Redmine by the product owner. This product backlog can be completed by specification or user requirements documents.
- The product itself which is the goal of the project.

Additional artefacts like timesheets or other project logs (meetings outcomes) can be useful or mandatory for reporting requirements according to project context.

4.4 Key Performance Indicators

Each team is responsible for maintaining and publishing a set of comparable Agile-compliant key performance indicators. Priority 1 indicators are obligatory. Priority 2 indicators are currently only highly recommended.

Key Performance Indicator	Type	Priority
Number of new bugs reported on Prod (Blocking/Critical) within last reference period (sprint, month)	Quality	1
Business user satisfaction: Collect with standard survey at team-specific times and reported quarterly	Quality	1
Average number of days for bug resolution on prod (Blocking/Critical) within last reference period (sprint, month)	SLA	1
Percentage of accepted stories by business (DoD) users vs committed stories by team within last reference period (sprint, month)	SLA	1
Average cycle time between when a story is delivered and the priority previously set by product owner (priority definition: a story has been sufficiently specified by business and has been marked to be done as soon as possible) for all stories delivered within the last reference period (sprint, month)	SLA	1
Average number of team members for the last reference period (sprint, month)	Volumetry	1
Man hours spent per project / lines of portfolio within the last reference period (sprint, month)	Volumetry	1
Unit test coverage percentage at the end of the last reference period (sprint, month)	Quality	2
Percentage hours spent per project within the last reference period (sprint, month)	Volumetry	2
Team happiness taken during each sprint retro	Motivation/Maturity	2
Number of re-opened bugs reported on Prod within the last reference period (sprint, month)	Quality	2

In addition, the team maintains a set of performance indicators for internal management purposes. These indicators are not supposed to be published but can be required to the team daily work. Some of them are classical scrum indicators, the others are custom. These indicators aren't necessarily comparable between different teams. Priority 1 indicators are obligatory. Priority 2 indicators are currently only highly recommended.

Indicator	Type	PRIORITY
Ratio between estimated story points and actual story point achieved per sprint	Maturity	1
Average number of story points achieved per person per sprint	Maturity	1
Sprint Velocity & Burndown chart	Maturity	1
Average number of days for code review	Quality	2
Average number of code review iteration	Quality	2
Number of code analysis rules activated without warning	Quality	2
Technical debt	Quality	2
Release burndown	SLA	2

All these indicators are derived from the artefacts of the projects. The implementation of the compilation of these different indicators is currently ongoing and correlated to the deployment of tools.

5. Common Toolbox

5.1 Backlog and management

Previously, several backlog management tools were coexisting at OECD. It has been decided to converge around a common tool set in order to capitalize on practices, knowledge and resources.

The selected target tool for operational backlog management is Redmine (<http://www.redmine.org/>).

The main OECD identified requirements covered by Redmine are:

- General agile project management (sprint, release)
- Efficient Bug/Task/Story/Epic management
- Customizable process
- Customizable Reporting
- Time tracking
- Source control integration
- Agile artifacts management (electronic boards, indicators)
- Open API
- Audit trail

The definition of the OECD project templates for Redmine and the definition of the required plugins have started. Each project team will have to evaluate the best opportunity of a migration and define the related actions plan.

As far as possible, Redmine will be used to collect the necessary information for an automated compilation of performance indicators.

5.2 Continuous integration

Continuous integration is not currently generalized in all OECD IT projects but this kind of approach is identified as a key enabler for industrialization of delivery processes.

Teamcity is identified as the target common tool for continuous integration (<https://www.jetbrains.com/teamcity/>).

Each project team is required to evaluate the opportunity and the feasibility of the implementation of Teamcity for continuous integration in their projects.

5.3 Other common tools

A target common tool will be identified for source code management. Currently, 3 code versioning systems are coexisting: Git, SVN and TFS.

A common technical toolbox will be defined in order to work on technical convergence when it's possible. Candidate tools are:

- Code quality analysis
- Web frontend frameworks
- Testing tools
- IDE